SYSTEM AND METHOD FOR SECURING NETWORK-CONNECTED RESOURCES

Invented by
Guy Eden
and
Lena Sojian

SYSTEM AND METHOD FOR SECURING NETWORK-CONNECTED RESOURCES

BACKGROUND OF THE INVENTION

1. Field of the Invention

This invention generally relates to encrypted communications and, more particularly, to a system and method for securing access to resources embedded in network-connected devices.

2. Description of the Related Art

There are situations in which a network administrator may seek to limit access to network-connected devices, such as printers, copiers, and multifunctional peripheral (MFP) devices. For example, if a printer is equipped with secure resources, such as font dual in-line memory modules (DIMMs), the fonts are vulnerable to theft or unauthorized use. Using basic hardware tools, a person can easily remove the secure font DIMM from the printer, and plug the DIMM on another printer, to gain access to the secure fonts.

One solution to this problem is to provide customers with a removable storage device to store the resource, in this case a secure font DIMM. This device houses the secure font DIMMS, and plugs directly into the printer when the fonts are needed. When the fonts are no longer needed, the device is unplugged from the printer, and stored for safekeeping. Although this solution provides some protection, it increases administrative overhead by making a person responsible for the secure font DIMM. This method also places the DIMMS at risk of being misused or misplaced.

It would be advantageous if device resources could be secured without having to physically remove the resources for safekeeping.

It would be advantageous if device resources could be encrypted in device memory and accessed using a cryptographic mechanism.

SUMMARY OF THE INVENTION

The present invention method secures device resources, such as fonts, by encrypting the resource before it is saved to DIMM. The encrypted fonts cannot be used until being decrypted using encryption keys. This provides a higher-level of security for storing secure printer fonts, and eliminates the added costs of maintaining extra hardware to secure the fonts.

Accordingly, a method is provided for securing network-connected resources. The method comprises: receiving an electronically formatted job at a first network-connected node; receiving CK, a symmetrical encryption key (K) encrypted using an asymmetrical encryption public key (pubK); and, receiving CH, a hash (H) of the job, further encrypted using K. Then, the method: decrypts CK using an asymmetrical encryption private key (privK), corresponding to pubK, to recover K; hashes the job, generating H'; uses K to validate CH; in response to validating CH, decrypts an encrypted resource using K; and, uses the decrypted resource to process the job.

In one aspect of the method, using K to validate CH includes: encrypting H' using K, obtaining CH'; and, matching CH to CH'.

Alternately, K is used to validate CH by: decrypting CH using K, generating H; and, comparing H to H'.

The received print job can be in either a text or an image format and, as mentioned above, the encrypted resource can be an encrypted font resource. Then, the print job can be printed using the decrypted fonts. The encrypted font resource can be a logo, personal signature image, or a glyph.

Additional details of the above-described method and a system for using secure network-connected resources are provided below.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic block diagram of the present invention system for using secure network-connected resources.

Fig. 2 is a schematic block diagram illustrating an alternate aspect of the system shown in Fig. 1.

Fig. 3 is a schematic block diagram illustrating a multi-device aspect of the present invention.

Fig. 4 is a schematic block diagram of the present invention system of Fig. 3, where multiple symmetrical encryption keys are used, in addition to multiple asymmetrical key sets.

Figs. 5a and 5b are flowcharts illustrating the present invention method for securing network-connected resources.

Fig. 6 is a flowchart illustrating the present invention method for accessing network-connected processing resources.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 is a schematic block diagram of the present invention system for using secure network-connected resources. The system 100 comprises a first device 102. The first device 102 includes a network-connected port on line 104 for receiving an electronically formatted job, and for receiving CK. CK is a symmetrical encryption key (K) encrypted using an asymmetrical encryption public key (pubK). Also received is CH, a hash (H) of the job, further encrypted using K.

A public key encryption algorithm (a.k.a.: asymmetric encryption) is an algorithm, which uses one key (a public key) for encrypting the message, and a second key (private key) for decrypting it. If Bob wants to send a ciphertext to Alice, he would use her public key for the task. While everyone can encrypt a message using Alice's public key, Alice is the only one who can decipher the message.

Symmetric encryption, also called conventional encryption, is any encryption system where the same key (K) is use for both encryption and decryption. This requires that the key must be securely transmitted between the encryptor and decryptor.

A one-way hash function typically takes a variable-length message and produces a fixed-length hash. It is computationally impossible to find the message in the hash. In fact, one can't determine any usable information about a message from its hash, not even a single bit. For some one-way hash functions, it's also computationally impossible to determine two messages that produce the same hash.

A hash unit 106 has an interface on line 104 to accept the job and an interface on line 108 to supply a hash of the job (H'). A memory 110 has an interface on line 112 to supply an asymmetrical encryption

private key (privK), corresponding to pubK, and an interface on line 113 to supply an encrypted resource (CR). A security unit 114 has an interface on line 116 to authorize access to the encrypted resource in memory 110, in response to validating CH. A processing unit 118 has an interface on line 104 to accept the job and an interface on line 120 to accept a decrypted resource (DR). The processing unit 118 has an interface on line 122 to supply a job processed using the decrypted resource. Although the processed job is shown as a paper media document, in other aspects of the system 100 (not shown) it is an electronically formatted document.

The system 100 further comprises a decrypting unit 124 having an interface on line 104 to accept CK and an interface on line 112 to accept privK. The decrypting unit 124 generates K in response to decrypting CK using privK. The decrypting unit 124 uses K to decrypt the encrypted resource from memory 110. The decrypted resource is supplied at an interface on line 120. An encryption unit 126 has an interface on line 108 to accept H' and an interface on line 121 to accept K. The encryption unit 126 supplies CH' at an interface on line 128 in response to using K to encrypt H'. The security unit 114 accepts CH on line 104 and CH' on line 128 and validates CH by matching CH to CH'. Thus, K must be derived (decrypted) from received information every time a secure resource is to be accessed.

Fig. 2 is a schematic block diagram illustrating an alternate aspect of the system shown in Fig. 1. The system of Fig. 2 is similar to the system of Fig. 1 except as noted below, and the similarities will not be repeated in the interest of brevity. In this aspect, the decrypting unit 124 has an interface on line 104 to accept CH and CK, as well as an interface

on line 112 to accept privK from the memory 110. The decryption unit 124 generates K, as in Fig. 1, by using privK to decrypt CK. Then, the decryption unit 124 supplies H on line 121 in response to decrypting CH using K. As above, the decryption unit 124 supplies the decrypted resource (DR) on line 120. The security unit accepts H on line 121 and H' on line 108, and validates CH by matching H to H'.

Referencing both Figs. 1 and 2, it should be understood that the system components are typically enabled as software, or microprocessor instruction sets. However, elements of the system may be enabled, or partially enabled, using hardware or firmware components. In one aspect of the system 100, the network-connected port on line 104 receives the encrypted resource for storage in the memory 110. That is, the encrypted resource need not necessarily be installed at the factory or during installation and initialization. The encrypted resource may be received in a hypertext transport protocol (http) or file transport protocol (FTP), for example. However, the invention is not limited to any particular format. To enhance the security of the system, the memory 110 (or a different memory, not shown) may be a read only memory (ROM) for accepting and storing privK upon device initialization.

In one aspect of the system, the first device 102 is a printer. As used herein, printer is understood to be an imaging device that is capable of generating a hardcopy document from an electronic document input. As such, the printer can be an MFP, scanner, or fax device. The invention is not limited to any particular document format. The network-connected port on line 104 may receive a print job in either a text format,

such as Word, or an image format, such as a portable document format (PDF) file.

If the first device 102 is a printer, then the encrypted resources in memory 110 may be encrypted font resources, and the processing unit 118 is a print engine that supplies a job on line 122 printed using the decrypted fonts. The encrypted font resources may be a logo, a personal signature image, or a glyph. For example, the personal signature image may be used to "sign" correspondence or checks. However, there are many types of symbols that can be protected for use by selected individuals.

In some aspects, the system 100 further comprises a second device 150, such as a network server or a personal computer. The second device 150 includes a processor 152 to supply the job on line 104. Note, the job may be supplied from memory or created by a document generation application. A hash unit 156 has an interface on line 104 to accept the job and an interface on line 154 to supply a hash of the job (H). An encryption unit 158 has an interface on line 154 to accept H, and an interface of line 104 to supply CK, the encryption of symmetrical encryption key K using pubK, and CH, the encryption of H using K. The second device 150 further includes a network-connected port on line 104 for transmitting the job, CK, and CH to the first device 102 for job processing.

As shown in Fig. 2, the first device network-connected port may receive an encrypted resource selection command on line 104. Then, the decryption unit 124 decrypts the selected resource (CR*i*). In this manner, numerous resources may be encrypted for use in a common

device. For example, different user groups may have differential access to the encrypted resources. More specifically, the decryption unit 124 receives and decrypts CK_i, where $1 \leq i \leq m$, to recover one of symmetrical encryption keys K_1 through K_m, where K_1 through K_m correspond to encrypted resources CR_1 through CR_m. Alternately stated, the particular K_i that is recovered in response to decryption CK_i, is used to decrypt a corresponding resource CR_i. Note, although not shown, this analysis applies to the system of Fig. 1, as well as the system of Fig. 2.

Fig. 3 is a schematic block diagram illustrating a multi-device aspect of the present invention. The system 300 comprises a plurality of devices N_i, where $1 \leq i \leq n$. The devices are similar to the first device described in the explanation of Figs. 1 and 2, and a detailed explanation will not be repeated here in the interest of brevity. Each device uses a different public/private asymmetrical key set. Shown are first device 102 and nth device 302. However, the system 300 is not limited to any particular number. Each device receives the electronically formatted job at a network-connected port on line 104, along with CK_i. In this aspect, CK_i is generated by encrypting K, using corresponding asymmetrical encryption public key $pubK_i$. Thus, first device 102 (N_1) receives CK_1, the encryption of K using $pubK_1$. Likewise, nth device 302 (N_n) receives CK_n, the encryption of K using $pubK_n$. Each device decryption unit decrypts CK_i using corresponding asymmetrical encryption private keys $privK_i$, to recover K. For simplicity, the same job is shown being sent to both devices 102 and 302. Practically however, the jobs are likely to be different, as they may be supplied from different user groups, or sent to different devices for alternate types of processing.

Fig. 4 is a schematic block diagram of the present invention system of Fig. 3, where multiple symmetrical encryption keys are used, in addition to multiple asymmetrical key sets. Again, each device N_i (where $1 \leq i \leq n$) receives the electronically formatted job at a network-connected port on line 104, along with CK_i. In this aspect, CK_i is generated by encrypting K_i using corresponding asymmetrical encryption public key $pubK_i$. For example, the first device 102 (N_1) receives CK_1, the encryption of K_1 using $pubK_1$. Each device also receives CH_i, a hash of the job encrypted using corresponding symmetrical encryption key K_i. For example, the first device 102 (N_1) receives CH_1, a hash of the job that is encrypted using K_1. Likewise, the nth device 302 (N_n) receives CK_n, the encryption of K_n using $pubK_n$, and CH_n, a hash of the job that is encrypted using K_n.

Each device decryption unit 124 decrypts CK_i using asymmetrical encryption private key $privK_i$, to recover corresponding symmetrical encryption key K_i. Then, K_i is used to decrypt of the encrypted resource CR. Thus, the first device 102 (N_1) decrypts CK_1 using $privK_1$, to recover K_1. K_1 is used to decrypt encrypted resource CR. Note, each device may store the same resource, different resources, or multiple resources. Again, for the sake of simplicity only, each device is shown receiving the same job. Typically, each device receives different jobs.

In one aspect of the invention, using the first device 102 as an example, the encryption unit 126 encrypts H' using symmetrical encryption key K_i, obtaining CH_i'. In this example, H' is encrypted using K_1, to obtain CH_1'. Then, the device security unit 114 validates CH by matching CH_i to corresponding CH_i'. In this example, CH_1 is matched to

CH_i'. A more detailed explanation of this validation process is provided in the description of Fig. 1.

In another aspect, using nth device 302 as an example, the decryption unit decrypts CH_i using symmetrical encryption keys K_i, obtaining H. In this example, H is obtained by decrypting CH_n using K_n. The security unit 114 validates CH by matching H to H'. A more detailed explanation of this validation process is provided in the description of Fig. 2. Note, the system depicted in Fig. 4 is not limited to the use of any particular CH validation method.

Functional Description

The present invention, enabled as a printer, may enact the following setup process:

1. The printer comes with a public/private encryption key (PrivK, PubK), which is setup at assembly time.

2. The administrator identifies the font as secure.

3. The administrator generates an encryption key K to protect the secure font.

4. The administrator uses K to encrypt the secure font, using a symmetric encryption algorithm. The administrator keeps the key used to encrypt the font (K).

5. The printer administrator uploads encrypted secure fonts to the printer using an upload mechanism provided by the printer manufacturer. This can be either FTP, HTTP, or any other network transport protocol.

6. The printer receives the secure font data and stores the font in its internal storage device. Note, K does not get stored on the printer and, thus, the printer can't decipher the font.

7. The administrator sends out K to all authorized users via a secure channel.

Following installation, the secure resource printer device may be used as follows:

1. Assume that an authorized user wants to send a print job and utilize the secure font.

2. The user encrypts K with the printer's public key (pubK) using an asymmetric algorithm, thus obtaining CK, which constitutes a cipher of K.

3. The user hashes the print job and obtains H, which is a hash of the print job.

4. The user encrypts the hash using a symmetric encryption and K as the key, and obtains CH.

5. The user sends the print job along with CK and CH.

6. The printer receives the print job, and recognizes it as referencing a secure font.

7. The printer attempts to recover K, which is the only way to decrypt and utilize the secure font.

8. The printer uses an asymmetric algorithm to decipher CK and compute K. It is guaranteed that the printer will succeed as it has the private key privK, corresponding to the public key pubK used to encrypt K. In fact, the printer is the only entity that can succeed in this task, as it is the only entity with knowledge of privK.

9. The printer hashes the print job and obtains H'.

10. The printer encrypts H' with a symmetric encryption algorithm, and K as the key, to obtain CH'.

11. The printer compares CH' with CH. If there is a match, then the printer can be confident that the user who sent the print job has legitimate access to K1 and, hence, is authorized to use the secure font. If CH' and CH do not match, the printer rejects the print job.

12. The printer uses K to decrypt the secure font previously uploaded by the administrator.

13. Once the printer computes the secure fonts, they can be utilized for the current print job. The printer uses a secure font to produce a print job.

14. The printer doesn't save a copy of the deciphered secure font, nor does it keep a copy of K, and so looses the ability to use the secure font again, until the next authorized print job arrives. The next authorized print job will reconvey K to the printer.

Note, the above-described utilization process corresponds to the aspect of the invention described by Fig. 1. The process described in Fig. 2 is similar, except for the specific CH validation method.

The following is a description of security provided by the present invention to possible attacks upon the secure resource.

The man in the middle attack:

1. Alice sends a print job to the printer, along with CK and CH.

2. Eve eavesdrops to the communication and intercepts CK and CH.

3. Eve's goal is to obtain K.

4. Eve has CK, which is the encryption of K. However, Eve cannot decipher CK without privK, the only way to decrypt CK.

5. Eve doesn't give up, even though the computation of K has failed. She still hopes to send her own print jobs and use the secure font.

6. Eve knows that CK never changes, and so she can add CK to her print job, which will be used by the printer to obtain K.

7. Eve knows how to compute H, which is the hash of her print job. But alas, what Eve cannot compute is CH, which is the encrypted hash of her document, using K as the key.

8. Thus, Eve cannot prove that she has legitimate access to K, and the printer rejects the print job.

9. The only possible attack that Eve can make is to record the whole session, and then impersonate to an authorized user, by sending the same print job as was previously sent by an authorized user. Then, CH matches the print job, and the print job won't get rejected. This attack is also known as a **replay attack.** However, this attack yields a very limited benefit to Eve, as she cannot author her own documents. In a sense, it is similar to producing a hard copy of a print job, and then making photocopies with a standard copier.

One strength of this invention is that the administrator can store multiple font sets, each requiring a different key to decrypt it (K_1, K_2,...K_n). This permits the administrator to set flexible rules as to what subset of users can use which fonts on the printer. In addition, the fonts can be copied to multiple printers. Each printer may have distinct public

and private keys ($pubK_1$,$privK_1$, $pubK_2$,$PrivK_2$,....$pubK_n$,$PrivK_n$) that may be used to enable the invention.

Furthermore, the key for decrypting the font is never stored on the printer itself, so no matter how far an attacker goes, they won't be able to utilize the font. The font cannot be decrypted even if the printer itself is stolen, and its innards hacked in a lab. Key distribution is a non-issue in many cases, as the administrator proliferates K to all authorized users. In a challenging environment, however, secure font keys proliferation is conducted via a public key encryption, in which every user has his own public-private key pair and, thus, the administrator can securely send K to authorized users.

Public encryption is relatively complex, on the order of 1000 to 1 more complex, as compared to symmetric encryption. If a printer had to decrypt print jobs, a bottleneck could easily develop. Therefore, instead of encrypting the print job, it is much cheaper (less computationally complex) to produce a hash of the print job, and encrypt the hash.

Figs. 5a and 5b are flowcharts illustrating the present invention method for securing network-connected resources. Although the method is depicted as a sequence of numbered steps for clarity, no order should be inferred from the numbering unless explicitly stated. It should be understood that some of these steps may be skipped, performed in parallel, or performed without the requirement of maintaining a strict order of sequence. The method starts at Step 500.

Step 502 receives an electronically formatted job at a first network-connected node. Step 502 can receive a print job in either a text or image format. Note that is some aspects of the invention, the input can

be a paper medium, such as blank checks requiring a (secure font) signature. However, this aspect still requires the use of an electronically formatted CK and CH, see Step 504 and 506. Step 504 receives CK, a symmetrical encryption key (K) encrypted using an asymmetrical encryption public key (pubK). Step 506 receives CH, a hash (H) of the job, further encrypted using K. Step 508 decrypts CK using an asymmetrical encryption private key (privK), corresponding to pubK, to recover K. Step 510 hashes the job, generating H'. Step 512 uses K to validate CH. Step 514 decrypts an encrypted resource using K in response to validating CH. Step 516 uses the decrypted resource to process the job.

In one aspect of the method, using K to validate CH in Step 512 includes substeps. Step 512a encrypts H' using K, obtaining CH'. Step 512b matches CH to CH'. Another aspect uses alternate substeps. Step 512c decrypts CH using K, generating H. Step 512d compares H to H'.

In one aspect, prior to receiving the job (Step 502), CK (Step 504), and CH (Step 506), Step 501a receives the encrypted resource. Step 501a may receive the encrypted resource in a format such as http or FTP. Step 501b stores the encrypted resource. For example, Step 501b may store an encrypted font resource. Then, using the decrypted resource to process the job in Step 516 includes printing a print job using the decrypted fonts. Step 501b may store resources such as a logo, personal signature image, or glyph. In another aspect, Step 501c installs pubK,privK upon initialization.

In one aspect, Step 501d generates the job at a second network-connected node. Step 501e encrypts K with pubK, generating

CK. Step 501f hashes the job, generating H. Step 501g encrypts H using K, generating CH. Step 501h sends the job, CK, and CH to the first node for job processing.

In one aspect of the method, a further step, Step 503, receives a selection command for a particular one of a plurality of encrypted resources. Then, decrypting an encrypted resource using K (Step 514) includes decrypting the selected resource. In another aspect, Step 503 receives a selection command for a particular one of a plurality of encrypted resources by receiving CK_i, where $1 \leq i \leq m$. In this aspect, Steps 503 and 504 are the same step. Then, decrypting the selected resource in response to the encrypted resource selection command (Step 514) includes decrypting CK_i to recover one of symmetrical encryption keys K_1 through K_m, where K_1 through K_m correspond to encrypted resources CR_1 through CR_m.

In another aspect, Step 502 receives the job at network-connected node N_i, where $1 \leq i \leq n$. Step 504 includes N_i receiving CK_i, where CK_i is generated by encrypting K using corresponding asymmetrical encryption public key $pubK_i$. Step 508 includes N_i decrypting CK_i using corresponding asymmetrical encryption private key $privK_i$, to recover K.

In a different aspect, Step 502 receives the job at network-connected node N_i, where $1 \leq i \leq n$, and Step 504 includes N_i receiving CK_i, corresponding to symmetrical encryption key K_i, encrypted using $pubK_i$. Likewise, Step 506 includes N_i receiving CH_i, a hash of the job encrypted using corresponding symmetrical encryption key K_i. Then, Step 508

includes N_i decrypting CK_i using asymmetrical encryption private key $privK_i$, to recover corresponding symmetrical encryption key K_i.

In Step 512a N_i encrypts H' using symmetrical encryption key K_i, obtaining CH_i', and in Step 512b N_i matches CH_i to corresponding CH_i'. Alternately, in Step 512c N_i decrypts CH_i using symmetrical encryption key K_i, obtaining H, and in Step 512d N_i compares H to H'. Either way, in Step 514 N_i decrypts the encrypted resource using symmetrical encryption key K_i.

Fig. 6 is a flowchart illustrating the present invention method for accessing network-connected processing resources. The method starts at Step 600. Step 602 generates an electronically formatted job at a second node. Step 604 encrypts a symmetrical encryption key K with an asymmetrical encryption key (pubK), generating CK. Step 606 hashes the job generating H. Step 608 encrypts H using K, generating CH. Step 610 sends the job, CK, and CH to a first network-connected node. Step 612 processes the job at the first node using a K encrypted resource.

A system and method for using encrypted network resources has been provided. The invention has been explained in the context of a printer loaded with encrypted fonts. However, the invention has broader application, to the secure use of any kind of network-accessible resource. Other variations and embodiments of the invention will occur to those skilled in the art.

WE CLAIM: